

---

## Steps to create a web application using Hibernate 3

---

**Step 1:** create a HTML page containing a **<form>** element.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Item Details</title>
</head>
<body>
    <form name="itemForm" action="ItemServlet" method="post">
        <table>
            <tr>
                <td>Item No</td>
                <td><input type="text" name="itemNo" /></td>
            </tr>
            <tr>
                <td>Item Name</td>
                <td><input type="text" name="itemName" /></td>
            </tr>
            <tr>
                <td>Quantity</td>
                <td><input type="text" name="quantity" /></td>
            </tr>
            <tr>
                <td>Unit Price</td>
                <td><input type="text" name="unitPrice" /></td>
            </tr>
            <tr>
                <td><input type="submit" name="option" value="Add" />
                <input type="submit" name="option" value="Update" /></td>
                <td><input type="submit" name="option" value="Delete" />
                <input type="submit" name="option" value="Show" /></td>
            </tr>
        </table>
    </form>
</body>
</html>
```

This is the name of the **servlet** file given in **@WebServlet**

**Step 2:** create a model / bean / POJO class.

```
package com.wipro.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
```

These annotations are used to map the Java class to DB Table

@Entity

@Table(name = "ItemDetails")

public class Item {

@Id

private int itemNo;

This annotation maps the **itemNo** as **PRIMARY KEY**

@Column(length = 15)

private String itemName;

This annotation is used to specify the column  
length, name etc.

private int quantity;

private float unitPrice;

public int getItemNo() {  
 return itemNo;  
}

public void setItemNo(int itemNo) {  
 this.itemNo = itemNo;  
}

public String getItemName() {  
 return itemName;  
}

public void setItemName(String itemName) {  
 this.itemName = itemName;  
}

public int getQuantity() {  
 return quantity;  
}

public void setQuantity(int quantity) {  
 this.quantity = quantity;  
}

public float getUnitPrice() {  
 return unitPrice;  
}

public void setUnitPrice(float unitPrice) {  
 this.unitPrice = unitPrice;  
}

}

**Step 3:** create a *HibernateUtil* class that returns a *SessionFactory* object.

```
package com.wipro.util;

import org.hibernate.cfg.Configuration;
import org.hibernate.SessionFactory;

public class HibernateUtil {
    private static SessionFactory sessionFactory;

    public static SessionFactory getSessionFactory() {
        try {
            sessionFactory = new
Configuration().configure("hibernate.cfg.xml").buildSessionFactory();
        } catch (Exception e) {
            System.out.println(e);
        }
    }

    return sessionFactory;
}
```

**Step 4:** create a *hibernate.cfg.xml* to configure hibernate parameters.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate
Configuration
    DTD 3.0//EN" "http://hibernate.sourceforge.net/hibernate-
configuration-3.0.dtd" -->
<!DOCTYPE hibernate-configuration SYSTEM
"classpath://org/hibernate/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <!-- Oracle dialect -->
        <property
name="hibernate.dialect">org.hibernate.dialect.OracleDialect</property
>

        <!-- Database connection settings -->
        <property
name="hibernate.connection.driver_class">oracle.jdbc.driver.OracleDriv
er</property>
        <property
name="hibernate.connection.url">jdbc:oracle:thin:@localhost:1521:XE</p
roperty>
        <property
name="hibernate.connection.username">system</property>
        <property
name="hibernate.connection.password">oracle</property>
        <!-- Echo all executed SQL to stdout -->
```

```

        <property name="hibernate.show_sql">true</property>
        <property name="hibernate.hbm2ddl.auto">update</property>
        <!-- Enable Hibernate's automatic session context management -->
        <property
name="hibernate.current_session_context_class">thread</property>

<mapping class="com.wipro.model.Item" />
    </session-factory>
</hibernate-configuration>

```

This name should be the bean class name

#### Step 5: *create a servlet to process the request.*

```

package com.wipro.servlet;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

import com.wipro.model.Item;
import com.wipro.util.HibernateUtil;

@WebServlet("/ItemServlet")
public class ItemServlet extends HttpServlet {

    private SessionFactory sessionFactory;
    private Session session;

    protected void doPost(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {

        //Retrieve all the parameters
        int itemNo = Integer.parseInt(request.getParameter("itemNo"));
        String itemName = request.getParameter("itemName");
        int quantity = Integer.parseInt(request.getParameter("quantity"));
        float unitPrice = Float.parseFloat(request.getParameter("unitPrice"));
        String option = request.getParameter("option");

        //Set the content type and get the writer

```

```

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        //Populate the object
        Item myItem = new Item();
        myItem.setItemNo(itemNo);
        myItem.setItemName(itemName);
        myItem.setQuantity(quantity);
        myItem.setUnitPrice(unitPrice);

        //Get the hibernate session from the factory
        sessionFactory = HibernateUtil.getSessionFactory();

        //Open the hibernate Session
        session = sessionFactory.openSession();

        //Begin the transaction
        session.getTransaction().begin();

        if (option.equals("Add")) {
            session.save(myItem);
            out.println("<h1>Data Saved Successfully!</h1>");
        }
        if (option.equals("Update")) {
            session.update(myItem);
            out.println("<h1>Data updated Successfully!</h1>");
        }
        if (option.equals("Delete")) {
            session.delete(myItem);
            out.println("<h1>Data deleted Successfully!</h1>");
        }
        if (option.equals("Show")) {

            List<Item> list = session.createCriteria(Item.class).list();

            out.print("<table border='1' bgcolor='yellow'>");
            out.print("<th>Item No.</th><th>Name</th><th>Quantity</th><th>Unit  
Price (Rs.)</th>");

            for (Item item : list) {
                out.print("<tr><td>" + item.getItemNo() + "</td><td>" +
                    item.getItemName() + "</td><td>" + item.getQuantity() + "</td><td>" +
                    item.getUnitPrice() + "</td></tr>");
            }

            out.print("</table>");
        }
    }

```

```

        //Commit the transaction
        session.getTransaction().commit();
    }
}

```

**Step 6:** Add the required jar files in **WEB-INF/lib** folder and Run the HTML file.

Item Details

localhost:8084/HibernateWeb/Item.html

Item No: 101  
Item Name: Pencil  
Quantity: 10  
Unit Price: 5.50  
Add Update Delete Show

localhost:8084/HibernateWeb/ItemServlet

Item No.	Name	Quantity	Unit Price(Rs.)
102	Pen	52	25.0
103	Stapler	25	30.0
105	Sketch Pens	200	35.0
101	Pencil	10	5.5

Item Details

localhost:8084/HibernateWeb/Item.html

Item No: 101  
Item Name: Pencil  
Quantity: 100  
Unit Price: 7  
Add Update Delete Show

localhost:8084/HibernateWeb/ItemServlet

**Data updated Successfully!**

localhost:8084/HibernateWeb/ItemServlet

Item No.	Name	Quantity	Unit Price(Rs.)
102	Pen	52	25.0
103	Stapler	25	30.0
105	Sketch Pens	200	35.0
101	Pencil	100	7.0

localhost:8084/HibernateWeb/ItemServlet

**Data deleted Successfully!**

localhost:8084/HibernateWeb/ItemServlet

Item No.	Name	Quantity	Unit Price(Rs.)
102	Pen	52	25.0
103	Stapler	25	30.0
105	Sketch Pens	200	35.0